

## Test di fine settimana – Week 4 - SQL

Nome	<u>Annamaria</u>
Cognome	<u>Ciasca</u>
Data	<u>06/08/21</u>

ATTENZIONE: Le domande a risposta multipla possono contenere più risposte corrette.

- 1) Ho 2 tabelle Tabella1 e Tabella2. Se scrivo **Tabella1 left join Tabella2** che risultati produce?
  - a) Tutti i valori presenti in Tabella1 e tutti quelli Presenti in Tabella2
  - ☒ b) Tutti i valori presenti in Tabella1 e tutti quelli in comune tra le 2 tabelle
  - c) Tutti i valori in comune tra le 2 tabelle.
  - d) Tutti i valori della Tabella2 e tutti quelli in comune tra le 2 tabelle
- 2) L'associazione "molti a molti" tra 2 tabelle come si traduce nel database? Inserendo un'ulteriore tabella che faccia da ponte fra le 2, contenente le foreign key relative ad entrambe
- 3) L'istruzione UPDATE è un comando di tipo:
  - a) Data Definition Language (**DDL**)
  - ☒ b) Data Manipulation Language (**DML**)
  - c) Data Control Language (**DCL**)
  - d) Transaction Control Language (**TCL**)
- 4) La funzione "SUM(attributo)" restituisce:
  - a) La somma degli attributi nella tabella
  - ☒ b) La somma dei valori posseduti dall'attributo
  - c) L'aggregazione dei valori dell'attributo
- 5) È possibile scrivere una condizione del tipo "where Id= null"?
  - a) Si
  - ☒ b) No      Bisogna usare IS o IS NOT

## Esercitazione pratica

Si vuole realizzare un sistema informativo per automatizzare la gestione di un negozio di dischi. In particolare il reparto degli Album delle Band.

Le entità coinvolte sono:

Band:

- Nome
- NumeroComponenti

Album:

- Titolo
- Anno di uscita
- Casa discografica
- Genere: può essere solo Classico, Jazz, Pop, Rock o Metal
- Supporto di distribuzione: può essere scelto tra CD, Vinile o Streaming

Brano:

- Titolo
- Durata (espressa in secondi)

È possibile che uno stesso brano faccia parte di più di un Album (ad es. le raccolte contengono brani appartenenti, in genere, ad album già pubblicati).

Una volta realizzato il **modello ER entità-relazionale**, creare il DB e tutte le tabelle e le relazioni necessarie.

Implementare oltre a quelli già esplicitati, anche i seguenti **vincoli**:

- Gli id devono essere auto-incrementati.
- Un album deve essere considerato unico sulla base del titolo, anno di uscita, casa discografica, genere e supporto (se uno stesso album viene memorizzato su, ad esempio, due supporti differenti, i dati relativi a quell'album devono essere registrati separatamente).

Realizzare le seguenti query SQL:

- 1) Scrivere una query che restituisca i titoli degli album degli "883" in ordine alfabetico;
- 2) Selezionare tutti gli album della casa discografica "Sony Music" relativi all'anno 2020;
- 3) Scrivere una query che restituisca tutti i titoli delle canzoni dei "Maneskin" appartenenti ad album pubblicati prima del 2019;
- 4) Individuare tutti gli album in cui è contenuta la canzone "Imagine";
- 5) Restituire il numero totale di canzoni eseguite dalla band "The Giornalisti";
- 6) Contare per ogni album, la "durata totale" cioè la somma dei secondi dei suoi brani
- 7) Mostrare i brani (distinti) degli "883" che durano più di 3 minuti (in alternativa usare i secondi quindi 180 s).
- 8) Mostrare tutte le Band il cui nome inizia per "M" e finisce per "n".
- 9) Mostrare il titolo dell'Album e di fianco un'etichetta che stabilisce che si tratta di un Album:  

'Very Old' se il suo anno di uscita è precedente al 1980,  
'New Entry' se l'anno di uscita è il 2021,  
'Recente' se il suo anno di uscita è compreso tra il 2010 e 2020,  
'Old' altrimenti.
- 10) Mostrare i brani non contenuti in nessun album.

Caricare la prova pratica e teorica su Github in un Nuovo Repository chiamato:

**NomeCognome\_ProvaQSL\_Week4.**